NPS-CAG-15-002

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

A STUDY OF COVERT COMMUNICATIONS IN SPACE PLATFORMS HOSTING GOVERNMENT PAYLOADS

by

Thuy D. Nguyen

February 2015

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE ***(DD-MM-YYYY)***
18-02-2015

2. REPORT TYPE
Technical Report

3. DATES COVERED ***(From-To)***

4. TITLE AND SUBTITLE

A STUDY OF COVERT COMMUNICATIONS IN SPACE PLATFORMS HOSTING GOVERNMENT PAYLOADS

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Thuy D. Nguyen

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) AND ADDRESS(ES)

Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

NPS-CAG-15-002

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited

13. SUPPLEMENTARY NOTES

The views expressed in this material are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

14. ABSTRACT

In recent years, unmanned space vehicles have increasingly become targets of cyber-attacks. Exacerbating the problem is the desire to reduce cost and accelerate access to space by hosting government-supplied payloads on commercial space platforms. These commercially hosted payloads require stringent confidentiality protection and encryption alone is not sufficient to protect against illegal information leakage on a spacecraft with multilevel security (cross-domain) capabilities. Covert channels may still exist and be exploited by colluding entities to communicate secretly via shared resources. This report describes a preliminary study of potential covert channels in communications protocols used in satellites—specifically MIL-STD-1553B and SpaceWire.

15. SUBJECT TERMS

Commercially hosted payload, covert channel, MIL-STD-1553B, SpaceWire, multilevel security, cross-domain

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			Thuy D. Nguyen
Unclassified	Unclassified	Unclassified	UU	53	**19b. TELEPHONE NUMBER** ***(include area code)*** (831) 656-3989

THIS PAGE INTENTIONALLY LEFT BLANK

NAVAL POSTGRADUATE SCHOOL
Monterey, California 93943-5000

The report entitled "*A Study of Covert Communications in Space Platforms Hosting Government Payloads*" was prepared for the Cyber Academic Group.

Further distribution of all or part of this report is authorized.

This report was prepared by:

Thuy D. Nguyen
Faculty Associate – Research
Department of Computer Science

Reviewed by:	**Released by:**
Cynthia E. Irvine Chair of Cyber Academic Group	Jeffrey D. Paduan Dean of Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

In recent years, unmanned space vehicles have increasingly become targets of cyber-attacks. Exacerbating the problem is the desire to reduce cost and accelerate access to space by hosting government-supplied payloads on commercial space platforms. These commercially hosted payloads require stringent confidentiality protection and encryption alone is not sufficient to protect against illegal information leakage on a spacecraft with multilevel security (cross-domain) capabilities. Covert channels may still exist and be exploited by colluding entities to communicate secretly via shared resources. This report describes a preliminary study of potential covert channels in communications protocols used in satellites—specifically MIL-STD-1553B and SpaceWire.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

In recent years, unmanned space vehicles have increasingly become targets of cyber-attacks. A 2011 report to Congress discussed several suspicious cyber events that interfered with two U.S. Government earth observation satellites in 2007 and 2008. The U.S. Geological Survey and National Aeronautics and Space Administration offices confirmed the attacks on the Landsat-7 and Terra EOS AM-1 satellites, respectively [1]. Exacerbating the problem is the desire to reduce cost and accelerate access to space by hosting government-supplied payloads on commercial space platforms.

The U.S. National Space Policy of 2010 mandates government organizations to leverage commercial space services to reduce the costs of government space missions, including the use of hosted government payloads on commercial spacecraft [2]. In its 2013 annual report, the Government Accountability Office (GAO) found that hosting government payloads on commercial satellite platforms has saved selected agencies hundreds of millions of dollars [3]. The GAO report highlighted a number of successful space projects that demonstrated the viability of the commercially hosted payload approach. For example, the Internet Protocol (IP) Routing in Space (IRIS) Joint Capability Technology Demonstration program was developed by an industry group in collaboration with the Department of Defense to provide IP routing onboard the commercial geostationary orbit satellite Intelsat 14 [4], and the Air Force's Commercially Hosted Infrared Payload (CHIRP) Flight Demonstration Program was an experiment that used a commercial communications satellite to host a classified infrared sensor payload [5]. Following the success of CHIRP, the U.S. Air Force started the Hosted Payload Solutions (HoPS) program to acquire on-orbit and ground services for government-furnished hosted payloads on commercial space platforms [6].

While hosting government payloads on non-government spacecraft promises both acquisition flexibility and cost savings, the security ramifications associated with the management of the satellite's shared resources with the satellite vendor and other payload owners require additional review and qualification. One of the shared resources used by the onboard systems is the intra-spacecraft data handling network that is shared among the spacecraft platform and its payloads.

To manage a shared network securely, both the temporal and spatial interactions among different modules residing on the network must be addressed. Without protection measures, one module could observe or corrupt another module's data, or misbehave and flood the network. Determining the conditions under which the confidentiality, integrity and availability of the hosted payloads can be maintained while sharing a data network is of critical significance.

A. MOTIVATION

For classified hosted payload, strong separation is required for information flow control in the presence of possible adversarial actions (e.g., malicious software co-resident on the commercial host). Threats to the commercial supply chain are just one motivation for considering appropriate isolation and protection of classified payloads on unclassified satellite hosts.

The *National Information Assurance Policy for Space Systems Used to Support National Security Missions* (*Committee on National Security Systems Policy No. 12*) requires the use of cryptography to encrypt all data transmitted over communications links, provide transmission security (TRANSEC) protection, and authenticate and encrypt all system commands [7]. However, regarding information flow control in a mandatory access control policy (i.e., no read-up and no write-down with respect to different levels of confidentiality), it has been shown that encryption is not sufficient to control all possible flows with different security classifications [8]. Covert and side channels may still exist in the presence of shared resources, e.g., the data buses between the classified payload and the space platform. A covert channel allows two cooperating entities to communicate secretly by manipulating shared resources, in violation of the security policy [9],[10]. In contrast, a side channel leaks information to other parties, and does not require the cooperation of some malicious entity on the high side.

Spacecraft communications architectures have evolved over the years. Conventional spacecraft typically use serial, circuit-based communications architectures, e.g., MIL-STD-1553B [11],[12], whereas advanced spacecraft gravitate towards network-based architectures, e.g., SpaceWire [13]. An undesirable effect of this technological progress is the increased risk for cyber attacks since complex technologies that require the dynamic allocation of resources across multiple components generally introduce emergent compositional vulnerabilities.

In the open literature, no prior work has empirically investigated attacks on MIL-STD-1553B (1553B herein) or SpaceWire, especially in terms of attacks disrupting information flow or non-interference properties—consequential for their use in hosted payload applications.

This report describes the initial findings of our study on covert channel attacks against communications protocols used in satellites with payloads operating at different sensitivity levels. Our work focuses on protocols defined in the 1553B and SpaceWire standards.

B. DEFINITIONS

For the purposes of this study, we adopted the definition of a covert channel from *A Guide to Understanding Covert Channel Analysis of Trusted Systems* [14], which states that a covert channel is "a communication channel that allows a process to transfer information in a manner that violates the system's security policy." Other definitions exist. For example, Schaefer et al. stated that a covert communication channel exists if it is based on "transmission by storage into variables that describe resource states" [15]. This definition, however, is specific to the storage of variables and its context is only meaningful for a stateful operating system. Kemmerer defines covert channels as those that "use entities not normally viewed as data objects to transfer information from one subject to another" [16]. While this definition is more generic and can be applied to stateless networks, it does not address the notion that covert channels depend on a system's security policy and how the system implements that policy. In this work, our use of the term 'side channel' is more closely aligned with Kemmerer's (policy-agnostic, unexpected) communication channels, whereas we reserve the term 'covert' to describe channels used by entities to violate an information flow policy.

We also adopted Kemmerer's definitions of storage channels and timing channels. Specifically, a covert channel is a storage channel if "the sending process alters a particular data item, and the receiving process detects and interprets the value of the altered data to receive information covertly." A covert channel is a timing channel if "the sending process modulates the amount of time required for the receiving process to perform a task or detect a change in an attribute, and the receiving process interprets this delay or lack of delay as information" [16].

C. REPORT ORGANIZATION

The remainder of this report begins with background information on intra-spacecraft communications, the 1553B bus architecture and the SpaceWire protocol suite. This is followed by a description of the threat models that provide the basis for our study. Next, we discuss the results of our preliminary analysis of potential covert channels in 1553B and SpaceWire protocols. Finally, we close with our conclusions and a brief description of future work.

II. BACKGROUND

Before discussing covert information flows in the context of intra-spacecraft communications, we review the onboard communications architecture of a satellite, the 1553B bus architecture, the SpaceWire network architecture, and two commercially hosted payload programs that were started by the United States Air Force.

A. SPACECRAFT C0MMUNICATIONS

A notional architecture of a satellite consists of an onboard computer, a number of subsystems, and one or more native and hosted payloads (see Figure 1 where the major subsystems are shaded.). The onboard computer controls the operation of the satellite, including commands execution, attitude and orbit control, time synchronization, failure detection and recovery, and housekeeping. The Communications subsystem manages the bidirectional communication channel between the satellite and ground station, and the Electrical Power subsystem controls the main power bus of the satellite. The Command and Data Handling (C&DH) subsystem handles commands and data sent and received by the spacecraft via the Communications subsystem.

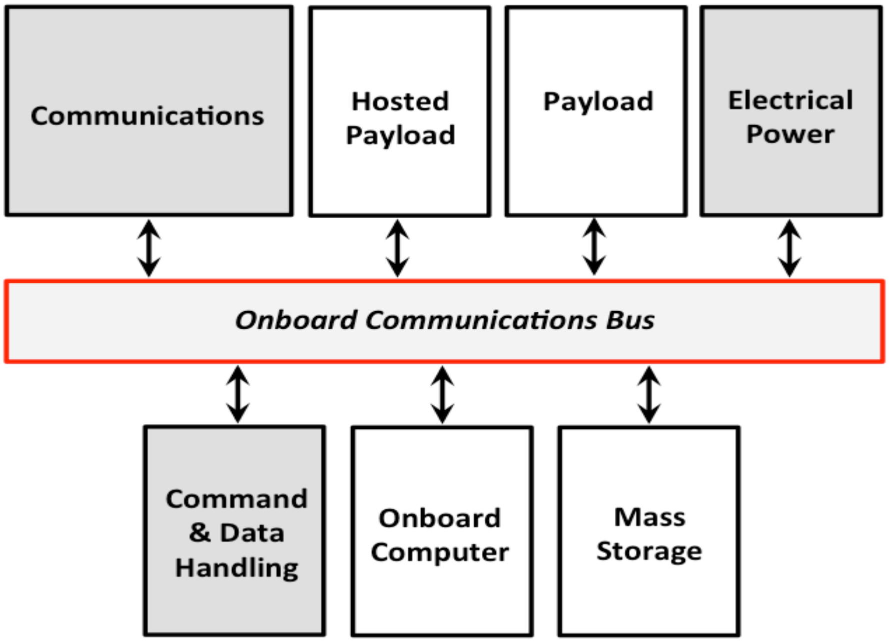

Figure 1. Notional spacecraft communications architecture.

On the uplink, the C&DH subsystem receives incoming commands and data for both the spacecraft platform and the payloads. It decodes the commands, and executes them if they are for the spacecraft platform. If the commands are for a payload, the C&DH subsystem forwards them to the target payload. The associated data is supposed to be passed to the payload unaltered and in a timely manner. On the downlink, the C&DH subsystem collects both spacecraft data and payload data, and transmits them to the ground segment.

B. MIL-STD-1553B ARCHITECTURE

The 1553B standard defines the physical and communications protocols of a shared, serial asynchronous data bus [11],[12]. The command/response protocols allow data exchanges between pairs of equipment connected to the shared bus. Functionally, a 1553B system is comprised of a data bus and a number of *terminals*. There are three types of terminals: *bus controller* (BC), *remote terminal* (RT), and *bus monitor* (BM). There are typically one BC, one or more RTs—up to 31 devices, and, optionally, one or more BMs. A 1553B subsystem is a functional unit of a terminal that sends or receives data from the data bus [17].

Bus management is accomplished via a strict master-slave relationship between the BC and RTs. The BC controls all data transfers on the bus. It instructs each RT on what bus operation to perform next, responds to service requests from individual RTs, and handles errors. A system can have more than one terminal that is capable of functioning as a BC, but only one terminal can operate as the BC at any given time. An RT performs specific operations in response to the BC's commands. The BM is a passive device that can monitor and record traffic on the bus.

The time division multiplexing, half-duplex command/response protocol defined in MIL-STD-1553B is commonly used in spacecraft for on-board data handling. All bus transmissions are accessible to all units connected to the bus, but only one unit can "speak" at a time. The BC initiates all bus transfers by sending a command message to a specific RT; the receiving RT is required to respond with a message acknowledging receipt of the BC's message.

A mission-critical system typically utilizes several 1553B buses to provide multiple data paths for redundancy [18],[19]. Figures 2 and 3 show two possible redundant bus configurations, the first with a single BC and the second with dual BCs, respectively.

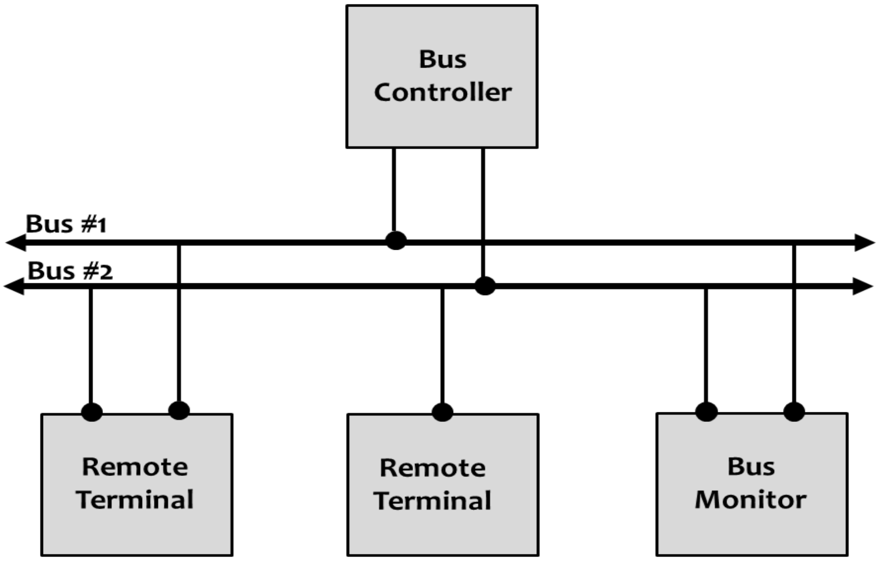

Figure 2. Dual redundant bus with single BC (after [11]).

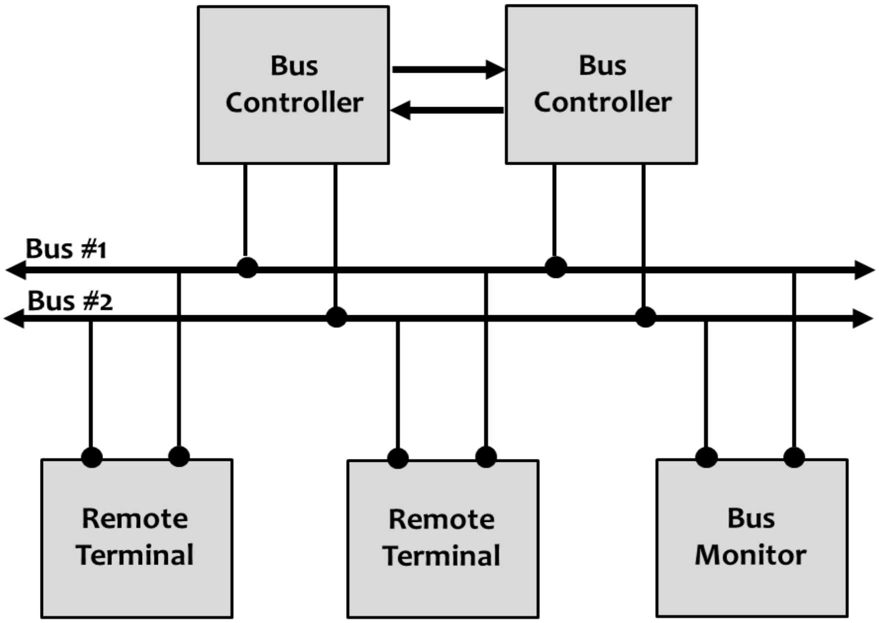

Figure 3. Dual redundant bus with dual BCs (after [11]).

For dual redundancy, the 1553B standard mandates that a *dual standby redundant data bus* architecture be used. In this configuration, only one data bus can be active at a time, except when a valid superseding command is sent on the standby bus [18]. The receiving terminal will abort the in-progress command if one exists and respond to the new command.

C. SPACEWIRE ARCHITECTURE

SpaceWire (SpW) is a spacecraft communications network that is based on the IEEE-1355-1995 standard for low latency serial interconnect [13]. The nodes in an SpW network are connected via bidirectional, full-duplex, point-to-point serial links, either directly or through packet switching routers. Figure 4 illustrates an example SpW network on a satellite with thirteen SpW nodes, two of which are routers.

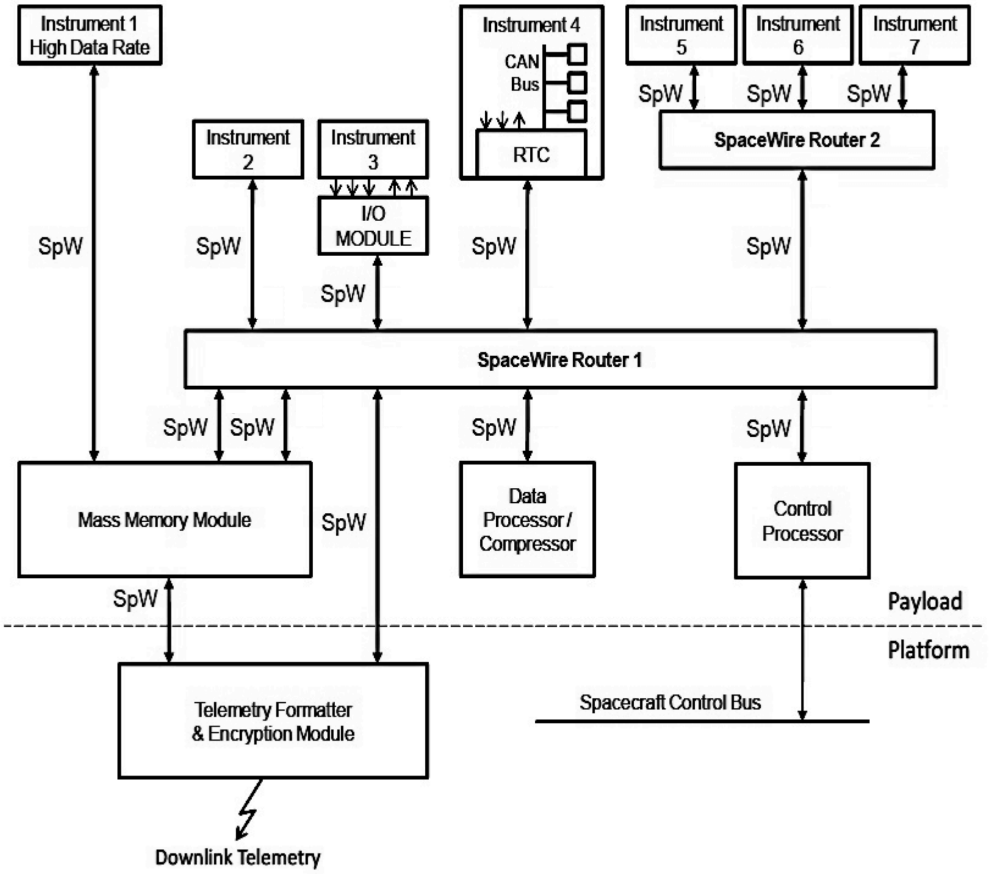

Figure 4. Example SpaceWire network (after [20]).

The SpaceWire standard defines six levels of protocols [13] that are summarized in ascending order below.

1. Physical level: This clause of the standard defines the mechanical and electrical interfaces between nodes, i.e., cabling, connectors, use of SpW over printed circuit boards and backplanes, and electromagnetic compatibility.
2. Signal level: This clause covers the signaling characteristics of an SpW link, i.e., voltage levels, noise margins, signal encoding, and data signaling rates.
3. Character level: This clause prescribes the character level protocol used to transfer data on an SpW link that includes parity detection, control characters, and a time interface to distribute the system time across an SpW network.
4. Exchange level: This clause describes the protocols used to initialize an SpW link, control the flow of data, detect disconnect and parity errors, and recover from a link error.
5. Packet level: This clause specifies the encapsulation scheme used to form data packets and the format of an SpW packet.
6. Network level: This clause defines the functionality of SpW nodes and routing switches (routers), the wormhole routing mechanism used by the routers to forward packets to the intended destinations, and network level error handling.

A non-routing SpW node is comprised of one or more SpW link interfaces [13]. Abstractly, each interface consists of a link receiver (input) and a link transmitter (output). Figure 5 shows the components of an SpW link interface and their allocation in the SpW protocol stack. A non-routing node does not include the network level protocols since they do not implement routing functions. An SpW time-code is a protocol data unit used to distribute system time over an SpW network.

An SpW routing switch (router) is composed of a number of SpaceWire link interfaces and a switch matrix [13]. The switch matrix (illustrated in Figure 6) uses the destination address of each packet to transfer packets from one link interface's input port to another link interface's output port.

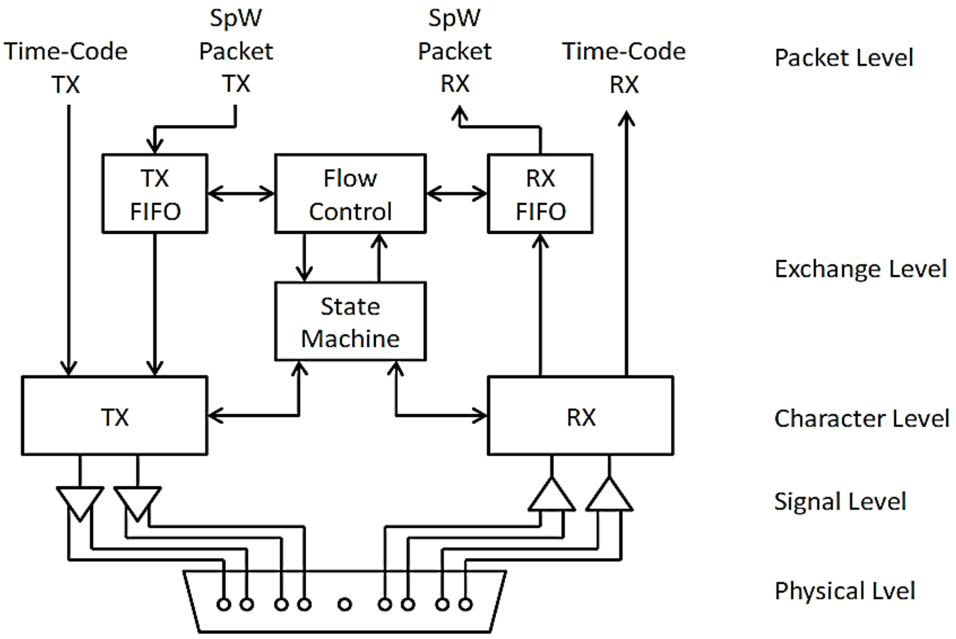

Figure 5. Block diagram of an SpW link interface [20].

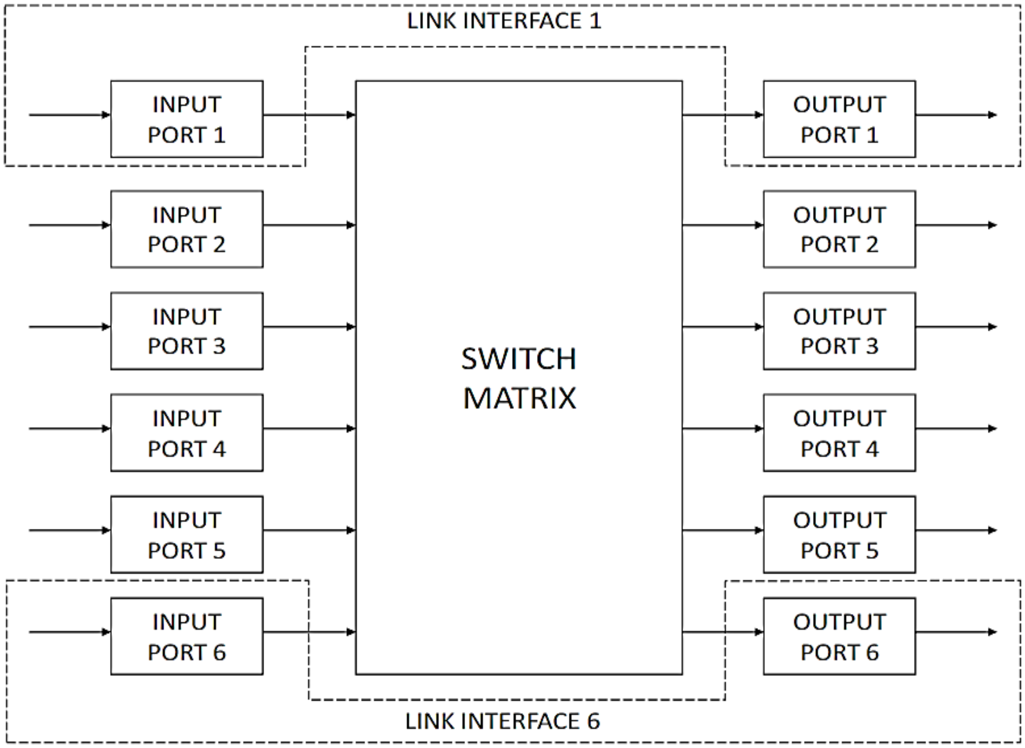

Figure 6. Block diagram of an SpW router [20].

D. COMMERCIALLY HOSTED PAYLOAD PROGRAMS

A hosted payload is traditionally developed together with the host space platform whereas a commercially hosted payload is designed and built without detailed knowledge of the hosting commercial spacecraft.

1. Commercially Hosted Infrared Payload Flight Demonstration

The Commercially Hosted Infrared Payload (CHIRP) Flight Demonstration (FD-CHIRP) program was started in 2008, launched in 2011, and successfully completed in 2012. It was the United States Air Force's first wide field-of-View staring infrared sensor flown in geostationary earth orbit (GEO) with a primary mission of missile warning that followed the commercially payload approach [21].

Except for sharing thermal and power with the space platform, the CHIRP payload operates with dedicated on-board processing subsystems payload and is isolated from the host platform. It is not connected to the host satellite's 1553B data bus and does not interact with the host satellite's Command & Data Handling (CD&H) unit or other subsystems. The CHIRP payload communicates with the government-owned ground system via a dedicated Radio Frequency (RF) link. It utilizes NSA Type 1 encryption for confidentiality protection.

During the launch of a space platform, i.e., between lift-off and when the space platform is released into orbit, the space platform communicates with the launch vehicle through a hardwired electrical connection to provide its state-of-health (SOH) telemetry during launch. Although the SOH telemetry is integrated and downlinked via the launch vehicle's data stream, the space platform remains passive and does not communicate with the ground system during launch. Similarly, the CHIRP hosted payload, and hosted payloads in general, are completely powered off during launch and ascent. Hosted payloads are not typically powered on until several weeks after launch once orbit-raising and initial satellite check-out is complete [22].

2. Hosted Payload Solutions

The HoPS program [6] specifies two reference mission architectures for protected hosted payloads: embedded and dedicated-link. For the embedded hosted payload architecture, the hosted payload transfers its commanding and telemetry streams via the space platform's commanding and data handling subsystem (right diagram in Figure 7).

For the dedicated-link architecture, the hosted payload transfers its commands and data through a dedicated transponder channel provided by the space platform (left diagram in Figure 7).

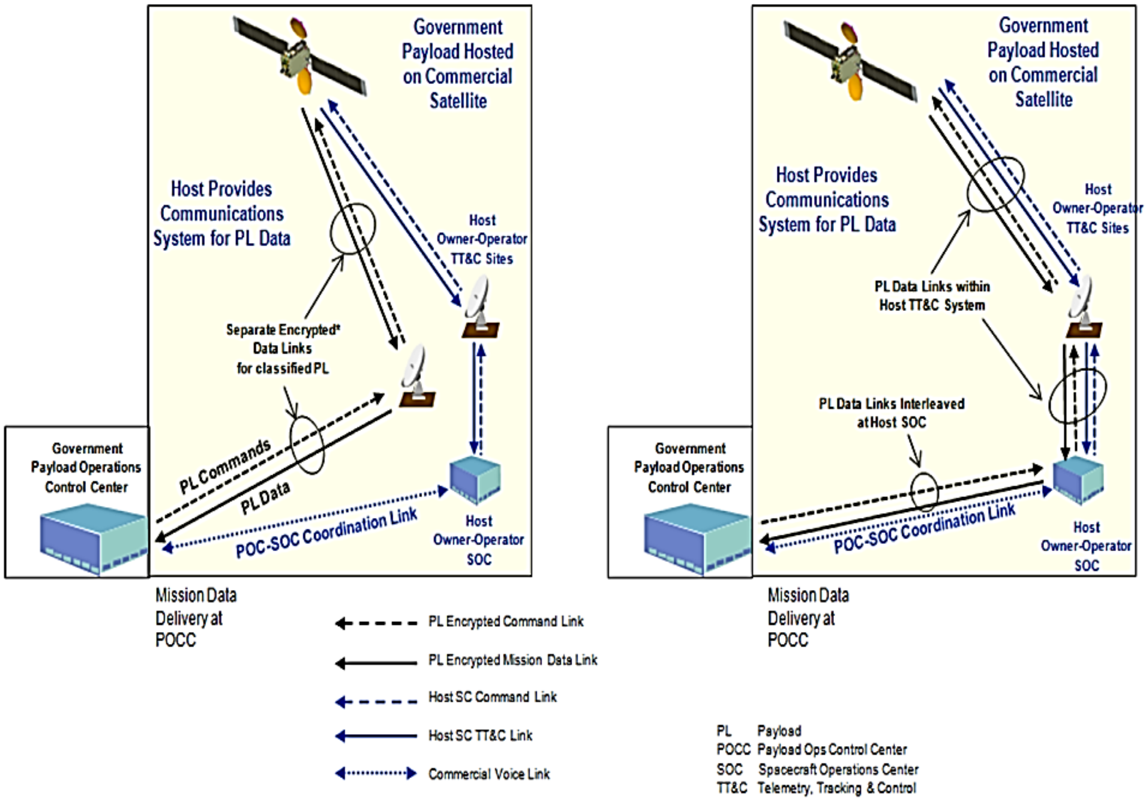
Government Payload Hosted on Commercial Satellite
Host Provides Communications System for PL Data
Host Owner-Operator TT&C Sites
Separate Encrypted* Data Links for classified PL
Government Payload Operations Control Center
PL Commands
PL Data
POC-SOC Coordination Link
Host Owner-Operator SOC
Mission Data Delivery at POCC
Government Payload Hosted on Commercial Satellite
Host Provides Communications System for PL Data
Host Owner-Operator TT&C Sites
PL Data Links within Host TT&C System
PL Data Links Interleaved at Host SOC
Government Payload Operations Control Center
POC-SOC Coordination Link
Host Owner-Operator SOC
Mission Data Delivery at POCC
PL Encrypted Command Link
PL Encrypted Mission Data Link
Host SC Command Link
Host SC TT&C Link
Commercial Voice Link
PL Payload
POCC Payload Ops Control Center
SOC Spacecraft Operations Center
TT&C Telemetry, Tracking & Control

Figure 7. Notional mission architectures for government payloads [23].

In both cases, a government-supplied Hosted Payload Interface Unit (HPIU) provides cryptographically-enforced separation between the protected hosted payload and the space platform (Figure 8). The HPIU supports MIL-STD-1553B, SpaceWire, RS-422, and Low Voltage Differential Signaling (LVDS) interfaces [23].

The NSA Type 1 crypto unit provides protection for the hosted payload’s commands and mission data. The payload’s analog telemetry data (e.g., power and temperature) are digitized and provided to the spacecraft’s C&DH subsystem to transfer to the ground system for safety monitoring purposes. Its high data rate allows the LVDS to be used to send data between the payload and the HPIU and between the HPIU and the crypto unit [23].

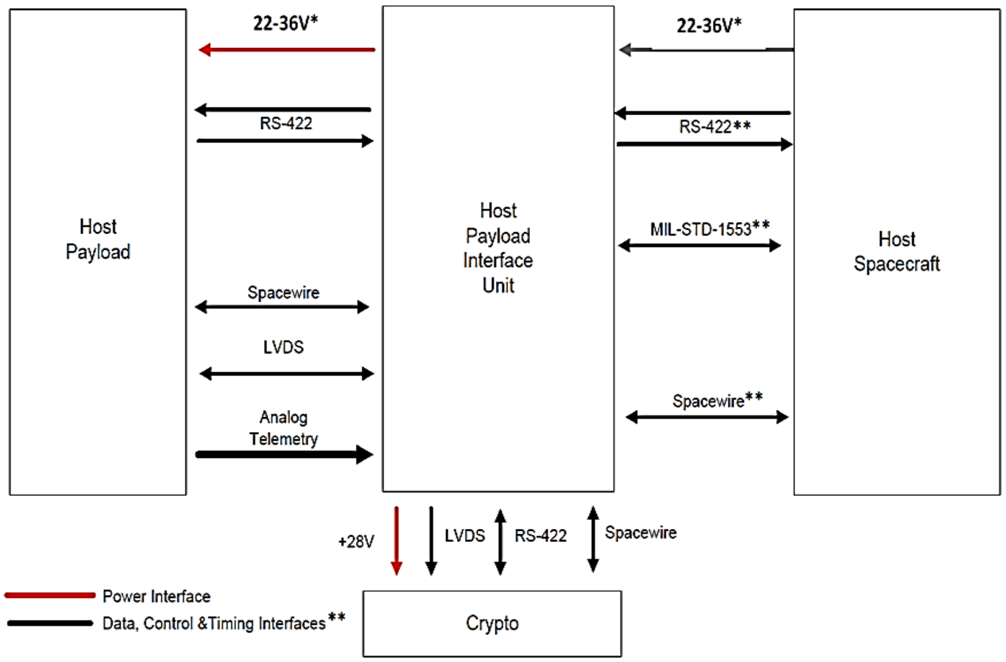

Figure 8. Notional Hosted Payload Interface Unit architecture [23].

The HoPS payload is required to undergo a certification and accreditation process which is to be identified during the requirement determination phase—either the cancelled *DOD Information Assurance Certification and Accreditation Process* (DIACAP) [24] or the recently-issued *Risk Management Framework for DoD Information Technology* [25]. The latter is based on the National Institute of Standards and Technology (NIST) Special Publication (SP) 800-37 (*Guide for Applying the Risk Management Framework to Federal Information Systems: A Security Life Cycle Approach*) [26] and NIST SP 800-53 (*Security and Privacy Controls for Federal Information Systems and Organizations*) [27].

The next chapter discusses the information assurance (IA) security controls defined by NIST SP 800-53 that are relevant to covert channel analysis.

THIS PAGE INTENTIONALLY LEFT BLANK

III. SECURITY CONTROLS

Annex H of *CNSS Instruction No. 1200* (*National Information Assurance Instruction for Space Systems Used to Support National Security Missions)* prescribes security requirements for space-based national security systems (NSS), including systems that use commercial space platforms to host NSS-payloads [28]. Of note are the requirements relating to: assured information flow control across security domains—cross domain solutions (CDS), separation of payload mission data from the host platform, payload command and control data processing, and information flows between host platform and payload's space and ground segments.

According to *DoD Instruction No. 8510.01*, a system must meet the IA security controls specified in the security overlays that apply to the specific system based on the information contained within the system or its operating environment [25]. Therefore, a space system with payloads operating at different sensitivity levels must meet the IA security controls specified in the *Space Platform Overlay* [29] and *Cross Domain Solution Overlay* [30]. The overlays are based on the following publications:

- NIST Special Publication 800-53, Revision 3, *Recommended Security Controls for Federal Information Systems and Organizations* [31];
- CNSS Instruction No. 1253, Version 2, *Security Categorization and Control Selection for National Security Systems* [32].

The Space Platform Overlay specifies a set of security controls required to address security risk to "unmanned space platforms in the space segment of national security space systems" [29]. The CDS Overlay defines the security controls required to protect systems that "provide access to and/or transfer of data between different security domains" [30].

With respect to covert channels, the CDS Overlay mandates covert channel analysis (CCA) to identify potential covert storage and timing channels in cross domain solutions. It also requires testing a subset of the identified covert channels to determine their exploitability. The Space Platform Overlay does not currently levy these controls on unmanned space platforms. Our analysis is intended to inform relevant concerns that have yet to be expressed in this domain, i.e., due to the lack of an overlay for hosted payload space applications.

Revision 4 of NIST SP 800-53 introduced two additional CCA control enhancements that require a system to 1) reduce the maximum bandwidth for identified covert channels and 2) measure the bandwidth of selected covert channels in the operational environment [27]. The Space Platform Overlay and CDS Overlay are being revised to conform to SP 800-53 Revision 4. It is anticipated that the updated Space Platform Overlay will include all CCA-related controls for space platforms that provide multilevel security (MLS) or cross domain capabilities.

NIST SP 800-53 Revision 4 defines several security controls relevant to covert channel analysis that are applicable to space platforms supporting MLS processing or hosting multiple payloads owned by different organizations. Table 1 summarizes these controls.

Table 1. Security controls relevant to covert channel analysis.

ID	Name
SA-11(6)	*Developer Security Testing and Evaluation | Attack Surface Reviews*
SA-15(5)	*Development Process, Standards, and Tools | Attack Surface Reduction*
SC-6	*Resource Availability*
SC-7(23)	*Boundary Protection | Disable Sender Feedback on Protocol Validation Failure*
SC-31	*Covert Channel Analysis*
SC-31(1)	*Covert Channel Analysis | Test Covert Channels for Exploitability*
SC-31(2)	*Covert Channel Analysis | Maximum Bandwidth*
SC-31(3)	*Covert Channel Analysis | Measure Bandwidth In Operational Environments*
SI-4(18)	*Information System Monitoring | Analyze Traffic / Covert Exfiltration*
SI-11	*Error Handling*

The SA-11(6) control enhancement requires the developer to perform attack surface reviews. Performing a covert channel analysis as part of vulnerability analysis and penetration testing is critical to the review process.

The SA-15(5) control enhancement requires the developer to reduce attack surfaces to specific thresholds. This enhancement is closely aligned with SA-11(6). CCA is an approach to measuring the attack surface and calculating the appropriate thresholds that are commensurate with acceptable risk.

The SC-6 security control requires the system to allocate critical resources according to priority, quota, or organization-defined service allotment to ensure their availability. Quota-based mechanisms can help mitigating timing channels (e.g., I/O scheduling) and storage channels (e.g., memory exhaustion) that could potentially exist when there are multiple payloads sharing a critical resource such as the onboard data bus. Priority-based mechanisms can ensure execution of high-priority mission-critical functions but they could cause covert channels to occur.

The SC-7(23) control enhancement requires the system to not provide feedback to senders when protocol validation fails. Many network covert channels can be implemented by manipulating protocol header fields in such a way to cause error messages [33].

The SC-31 security control requires a CCA be performed to identify potential storage and timing channels and to estimate the maximum bandwidth of those channels. This control has three control enhancements. The SC-31(1) enhancement confirms the exploitability of a subset of identified channels by testing. The determination of which channels to test is typically based on the ease of exploitation and the value of the leaked information. The SC-31(2) enhancement requires the developer to reduce the maximum bandwidth of selected channels to predefined limits deemed acceptable. The bandwidth values are also calculated based on the ease of exploitation and the value of the leaked information. The SC-32(3) enhancement requires measuring the bandwidth of a subset of identified channels in the environment in which the system operates. The bandwidth of a covert channel may vary when measured in different environments with different operating settings, e.g., operational versus development testbed.

The SI-4(18) control enhancement requires an analysis of covert exfiltration of information at the system perimeter and at predefined interior access points within the system. Some covert exfiltration techniques are similar to covert channel techniques, e.g., manipulation of protocol headers or exploitation of protocol response time, the identified covert channels may be useful to the covert exfiltration analysis.

The SI-11 security control requires the system to exclude information that could be exploited by adversaries in error messages. Error messages may be used as a signaling mechanism to covertly transfer information, e.g., an error message describing a resource exhaustion condition can be manipulated to transmit zeros or ones.

The selection of these CCA-related security controls and control enhancements for a space system with MLS capabilities will be based on the potential adverse impact on the particular system and its assets if the system was compromised [27].

Next, we discuss the threat model we use to identify potential covert channels in the 1553B and SpaceWire protocols.

IV. THREAT MODEL

For our work, the threat model is that of two colluding adversarial entities that communicate covertly, in violation of the system security policy, to leak information that is otherwise not available to unprivileged entities. For our analysis, we apply the traditional MLS confidentiality policy as modeled by Bell and LaPadula [34]. This policy disallows the flow of information from a unit operating at a high level of confidentiality and a unit operating at a low level of confidentiality.

The 1553B standards specify three types of validation testing that a 1553B unit must pass to ensure conformance to the 1553B specification: electrical, protocol, and noise reduction [11]. The SpaceWire specification also includes a set of criteria to which an SpW implementation can claim to conform if it satisfies the implementation-specific criteria [13]. Passing the 1553B validation tests and meeting the SpW conformance criteria do not guarantee that the design and implementation of a 1553B or SpW unit is trustworthy or free of unintended leakage channels. Furthermore, adversaries can subvert software during the development process and supply chain to either create or exploit channel vulnerabilities.

For government hosted payload missions, the critical payload data are encrypted. However, protocol metadata, e.g., information in packet headers, are transmitted in the clear on the shared communications channels. Our covert channel analysis focuses on attacks that utilize this type of data. We began by identifying salient 1553B and SpaceWire features that can potentially be exploited through observation of: 1) timing behavior, 2) component-specific implementation features, and 3) protocol control information using valid protocol operations.

Channels that require persistent physical access to exploit (e.g., channels exploitable via differential power analysis) are out of scope of our analysis. For remote space platforms, it may be unrealistic exploit such channels they are generally difficult to orchestrate via malicious software or firmware.

A. 1553B ATTACK SCENARIOS

As applied to the 1553B protocol, the applicable security policy is a traditional mandatory access control policy in which the active entities with the potential to cause information flow, viz. subjects, map to the units connected to the 1553B bus, and "information" maps to *data words* defined in the 1553B protocol. The data transmitted via 1553B data words are specifically distinguishable from other information defined in the 1553B protocol in that the contents of data words are not interpreted by the 1553B protocol in any way, i.e., have no semantic meaning to the protocol.

This may be contrasted with other aspects of the 1553B protocol such as *command words* and *status words* that define the formats and semantically meaningful control information that form the basis of the protocol. In this context, "covert channel" refers to the illegal flow of information effected via 1553B protocol constructs.

In Figure 9, we show a simplified 1553B configuration used to construct the hypothesized 1553B attack scenarios.

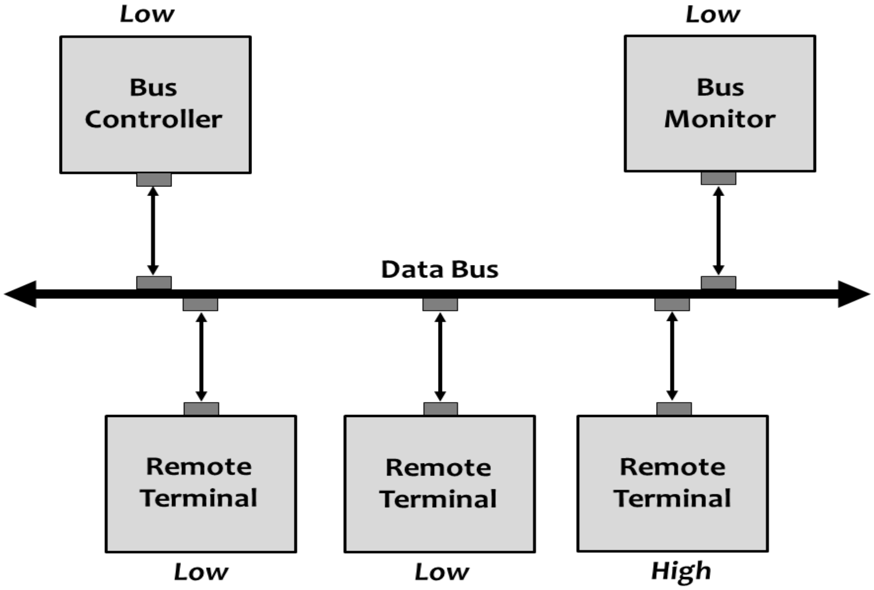

Figure 9. A 1553B configuration with terminals operating at different sensitivity levels.

We formulate the attack scenarios described in Table 2 based on the 1553B protocol description and 1553B product manuals. For these attacks, the exploitable "protocol control information" refers to any 1553B-defined control and status fields;

1553B-defined data payload fields are specifically excluded. Since the BC controls the bus, our study does not include attack scenarios in which the BC acts maliciously.

Table 2. 1553B attack scenarios.

Type	Attack Scenarios
Timing attack	*Observation of RT-specific timing behavior*
Storage attack (#1)	*Observation of RT-specific protocol control information using RT implementation-specific features*
Storage attack (#2)	*Observation of protocol control information using valid protocol operations*

Construction of timing channel attacks does not distinguish the use of valid protocol operations versus invalid and incorrect protocol operations since it is assumed all timing behavior of a specific RT is unique. For the identified timing channel, the two cooperating RTs (a Low RT and the High RT) use RT-specific timing behavior to transmit and receive High information. This scenario is discussed in Section V.A.1.

The first storage channel attack involves two cooperating RTs and the High RT implements an optional function, which the two RTs agree *a priori* to use as the signaling mechanism. This scenario is discussed in Section V.A.2.

In the second storage channel attack, the two cooperating RTs only use regular protocol operations that are supported by all 1553B terminals. The Low RT can extract High information by observing the way the High RT uses a particular operation. This scenario is discussed in Section V.A.3.

B. SPACEWIRE ATTACK SCENARIOS

In contrast to a multidrop shared bus such as 1553B, SpaceWire is a switched network with nodes connected via point-to-point links. Each SpW node may have one or more SpW interfaces and, in the commercially hosted payload context, the interfaces in a multi-interface (MI) node can operate at different sensitivity levels. Figure 10 illustrates this configuration. Due to the direct connection between nodes, malicious disclosure between two colluding nodes attached to the same MI node must be accomplished through their use of shared resources in the shared node. An MI node can be a routing node or a non-routing node, e.g., a mass storage device.

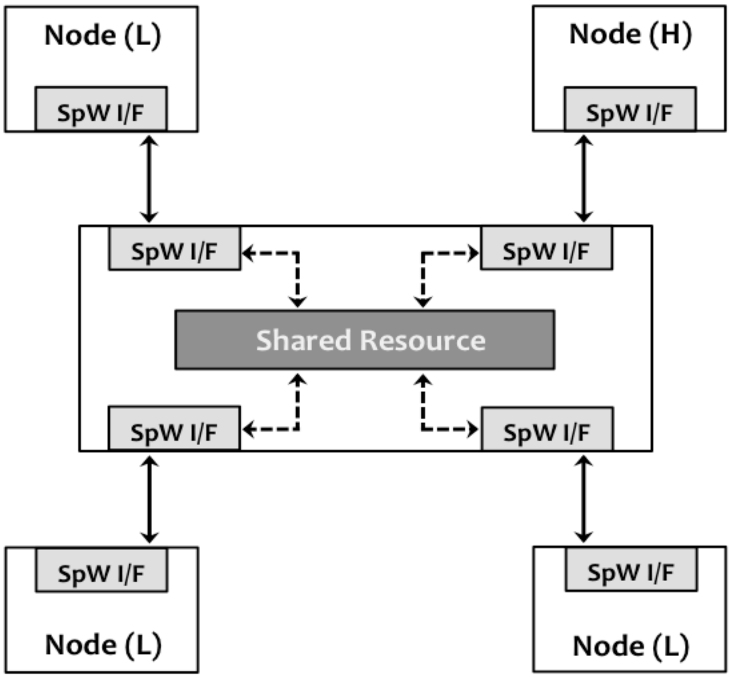

Figure 10. A SpaceWire configuration with nodes operating at different sensitivity levels.

A timing covert channel can potentially exist in an MI node if a High node could modulate its use of the MI node in a way that a Low node could detect a change in the amount of time required for the Low node to access a shared resource on the MI node, e.g., memory in a mass memory device. Unless the MI node partitions the shared resource and gives its SpW interfaces their own memory chunks, a storage covert channel can potentially be constructed if the shared resource manager on the MI node returns an error message (e.g., resource exhaustion) that the connected SpW nodes can observe.

Analyzing covert communications at the three lowest protocol levels of SpaceWire (physical, signal and character) is beyond the scope of our analysis. The next chapter discusses ways to exploit these timing and storage covert channels.

V. PRELIMINARY ANALYSIS

The first step for managing covert channels in a system is to identify them either in an informal way or based on a formal method, e.g., information flow analysis [35], Shared Resource Matrix methodology [36] and non-interference analysis [37]. Our identification process is *ad hoc* and loosely based on the *flaw hypothesis methodology* [38],[39],[40], i.e., identifying potential covert channels through analysis of protocol specifications and documentation of various products.

A. 1553B COVERT CHANNELS

Design and implementation choices that could potentially be exploited in a 1553B system include: 1) out-of-spec exception handling, such as the incorrect use of the subaddress field for data wraparound; 2) undefined behavior, such as how the BC and RTs handle undefined error conditions, optional parameters and optional commands; and 3) bus control and monitoring—for example, considering acyclic data transfer vs. scheduled data transfer by RTs, or considering recording-only (passive) bus monitors vs. hybrid bus monitors (i.e., able to serve as a back-up bus controller). These conditions provide fodder for the construction of the storage and timing covert channels summarized in Table 2 and described below. A more complete description of these attacks is provided in a separate report [41].

1. RT response time delay

This timing covert channel allows two cooperating RTs running at different security levels to use the RT response time delay to transmit and observe information. In a 1553B system, the BC initiates all commands and each RT must respond to a valid command within a time period of 4 to 12 microseconds. Since every connected RT can observe all transmissions on the data bus, any RT can identify all commands sent by the BC to another RT and all responses returned by the target RT. Hence, a Low RT (receiver) could detect the covertly transmitted information by monitoring the response time delay introduced by a malicious High RT (sender). The signaling mechanism would be the amount of time delayed by the High RT before responding to a command. Figure

11 illustrates an example 1-bit timing channel in which the High RT varies the response time to send different bit values.

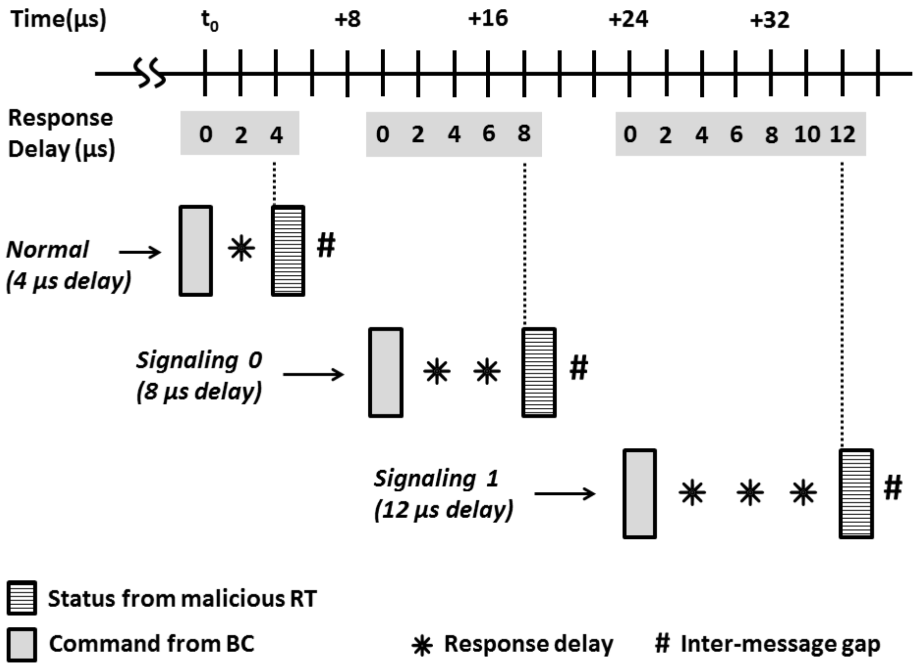

Figure 11. RT response time delay timing channel.

A Low RT can receive up to three bits of information per message if the High RT could control the response time delay to a granularity of one microsecond. Several 1553B products provide board-level programming interfaces that allow applications to set the granularity of the RT response time down to nanoseconds. For example, the AIM PCI 1553 bus interface module allows the RT response time to be programmed in 250 nanosecond increments [42]; the AltaCore-1553 protocol engine supports a 12-bit RT response time with a resolution of 100 nanoseconds [43]; and the Excalibur multifunction interface module allows the RT response time to be set within the range of 4000 – 42000 nanoseconds [44].

A typical timing channel requires the sender and receiver to have either a common clock or the ability to create a time reference [14]. For this timing channel, the colluding RTs can use the end of a BC command as the common time reference. Synchronization between the two RTs can start after a command word is detected; a command word is always preceded by an inter-message gap of at least 4 microseconds (introduced by the BC), and consists of: a 3-bit sync pattern, sixteen bits of information, and one parity bit.

2. Command illegalization

This storage channel allows two cooperating RTs to use RT-specific implementation of the “command illegalization” function to leak information. An illegal command is a valid command that is not in the set of commands specified for use with the target RT. Depending on its design, an RT may treat certain commands as illegal and will return a status word with the message error (ME) bit set when such commands are detected. A BC can automatically retry the issued command when it receives a status word response for that command with the ME bit set.

Our analysis assumes that the BC implements automatic retrying and the High RT supports command illegalization. The BC must retry at least once on the same bus and the High RT must be able to dynamically reject an arbitrary command as illegal. Furthermore, the High RT and Low RT must previously agree on when the signaling method will be utilized. For example, the agreement can be based on a specific command or commands sent from the BC. Whenever the BC sends a specific command to the High RT, the High RT will signal a bit by selectively rejecting the command or accepting the command. The Low RT observes the High RT’s action to receive the transmitted bit. The sequence chart in Figure 12 shows this signaling mechanism.

An RT that supports command illegalization typically provides a programming interface for the RT’s subsystem to configure the accepted illegal commands which are kept in an *illegalization database* in either register or memory. When used as an RT, both the Microsemi Core1553BRM [45] and Aeroflex SμMMIT [46] protocol engines support RT command illegalization.

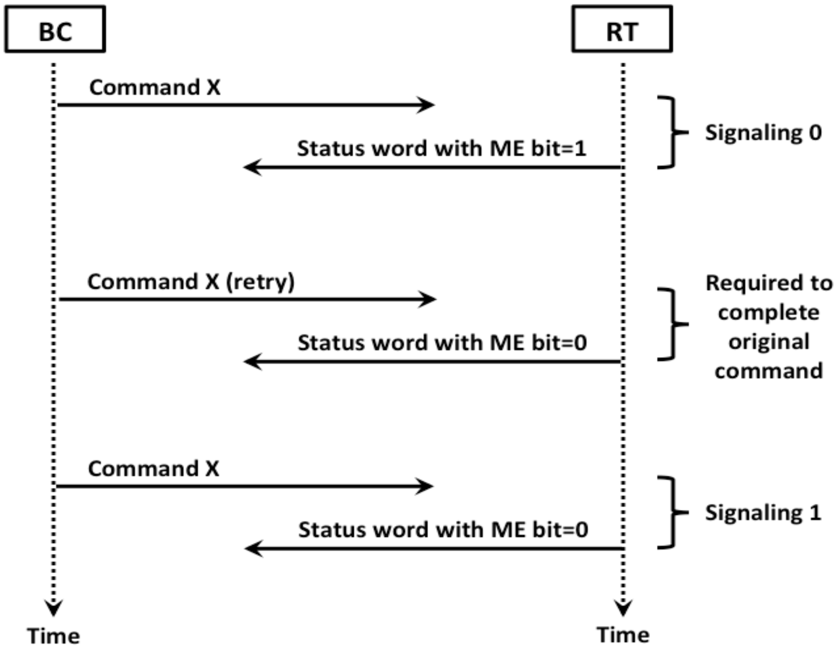
BC
RT
Command X
Status word with ME bit=1
Signaling 0
Command X (retry)
Status word with ME bit=0
Required to complete original command
Command X
Status word with ME bit=0
Signaling 1
Time
Time

Figure 12. RT command illegalization storage channel.

3. Acyclic transfer

This storage channel enables two cooperating RTs to leak information using the Service Request (SR) bit in the status word. In a 1553B system, the BC issues commands to the RTs in a cyclic sequence. When an RT wants to request an acyclic (asynchronous) data transfer, it returns a status word with the Service Request (SR) bit set. When the BC detects the Service Requested condition, the BC either executes a predefined function or sends a Transmit Vector Word command to obtain additional information from the RT about the requested service.

In our analysis, we hypothesize that the BC responds to an asynchronous service request by issuing the Transmit Vector Word mode command, after which the RT responds by transmitting a vector word. The Low RT receives bits from the High RT by observing the way the High RT issues an asynchronous request. Figure 13 illustrates a potential channel.

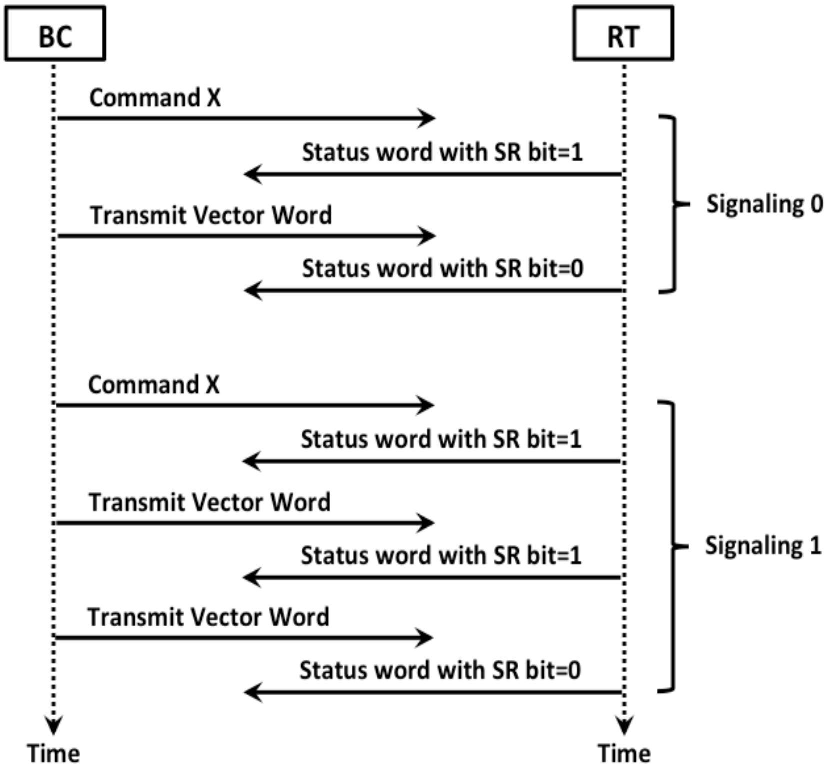
BC
RT
Command X
Status word with SR bit=1
Transmit Vector Word
Status word with SR bit=0
Signaling 0
Command X
Status word with SR bit=1
Transmit Vector Word
Status word with SR bit=1
Transmit Vector Word
Status word with SR bit=0
Signaling 1
Time
Time

Figure 13. Acyclic transfer storage channel.

If the High RT follows up the request with transmission of a non-zero vector word, the Low RT knows the High RT is merely performing a legitimate asynchronous request action. If the High RT transmits a single empty vector word, the Low RT interprets the High RT's action as transmission of a bit value of 0. If the High RT transmits two consecutive empty vector words, the Low RT interprets the High RT's action as transmission of a bit value of 1.

All 1553B products that we survey support the use of the SR bit to request asynchronous service.

B. SPACEWIRE COVERT CHANNELS

This section describes several covert channels that may potentially exist in an SpW mass memory device and an SpW router.

1. Mass memory device scenarios

A mass memory device (MMD) on a spacecraft (shown in Figure 4) is shared by multiple components, either directly or indirectly via a router. Its primary function is to provide memory-based storage that other equipment on the spacecraft (e.g. sensors, scientific instruments, etc.) to save their data until the data can be transmitted to the ground system.

A representative design of the MMD used in our study is modeled after the INTAμSAT's mass memory unit reported by Garcia, et al. [47] (see Figure 14). The MMD has multiple SpW link interfaces, each is connected to an external SpW node. The MMD is itself an SpW node in the SpW network. Internally, the memory device in each SpW link interface (not shown in Figure 4) communicates with the SDRAM controller via an Advanced Microcontroller Bus Architecture (AMBA) Advanced High-Performance Bus (AHB) bus. The SDRAM controller manages the external SDRAM chips and the on-chip processor interacts with the spacecraft's Telemetry (TM) and Telecommand (TC) system. The processor also manages the SpW link interfaces via the AMBA AHB bus.

AMBA AHB is a high-performance master-slave system bus. An AMBA AHB system typically includes one or more bus masters, one or more bus slaves, a bus arbiter, and a bus decoder [48].

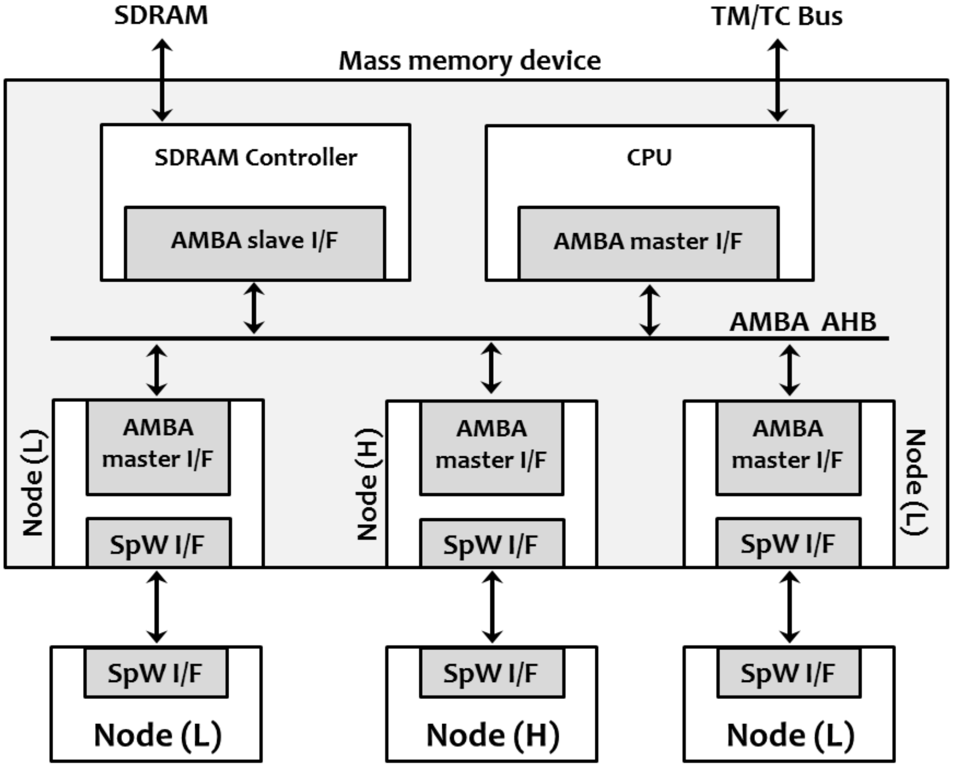

Figure 14. Mass memory device block diagram (after [44]).

A bus master can initiate read and write operations; a bus slave performs the read and write operations initiated by a bus master; the bus arbiter decides which bus master is allowed to initiate data transfer on the bus; and the bus decoder performs address decoding of each transfer and selects the slave involved in the data transfer. The bus arbiter ensures that only one bus master is active on the bus at a given time.

Once a data transfer is started, the initiating bus master cannot cancel that transfer. During a transfer, a slave can return one of the following responses: OKAY, ERROR, RETRY and SPLIT. The OKAY response indicates that transfer is either progressing normally or has completed successfully. The ERROR response tells the master that transfer has terminated unsuccessfully due to an error. Both the RETRY and SPLIT responses indicate that the slave cannot complete the transfer immediately and has released the bus so that another bus master can gain access to the bus. With RETRY, only a master with higher priority can use the bus whereas with SPLIT, any bus master requesting the bus can get access regardless of its priority. The original bus master is expected to retry the transfer until it either completes successfully or terminates with an ERROR response.

Several properties of the MMD can be exploited to create a covert channel:

1. The MMD does not fully buffer or otherwise virtualize data transfer requests, e.g., it does not present a non-blocking data transfer interface to all attached nodes.
2. The MMD implements bidirectional flow of control information between the MMU and all connected nodes. Control information can include commands, error and status information.

As a shared resource that presents a non-virtualized interface, the MMD is definitely the source of a covert timing channel. A storage channel may also exist, depending on the control information made available by the MMD.

Abstractly, a timing covert channel is created by a malicious High node cooperating with a malicious Low node to time their requests to the MDD node in a manner such that contention to the shared MMD node can be detected.

The MMD design utilizes the AMBA bus to connect multiple SpW links, which in turn connect to multiple devices that read and write data to the MMD. Within this

MMD design, the AMBA bus is the shared resource whose use is modulated to create the covert channel.

Only one AMBA bus master at a time may perform a transfer. If a second bus master requests a transfer while the bus is in use by the first bus master, the second bus master must wait until the current transfer is complete before the second bus master may access the bus. Malicious nodes exploit this behavior to transmit information by timing MMD data transfer requests to either create contention for the AMBA bus (a 1 value) on not create contention for the AMBA bus (a 0 value).

First the Low node establishes a baseline for how long an MMD data transfer should take when there is no contention on the AMBA bus, i.e., there is no other active MMD data transfer.

To transmit a 1 value, the High node requests an MMD data transfer at Time 0. The Low node will request an MMD data transfer at Time 1 and measure the time it takes to complete the MMD data transfer. If the data transfer takes longer than the baseline value, the Low node will know that the High node has transmitted a 1 value by initiating an MMD data transfer prior to the Low node data transfer request.

To transmit a 0 value, the High node will take no action at Time 0. The Low node will request an MMD data transfer that will complete within the baseline time because there was no contention for the AMBA bus, thus a 0 value is signaled from the High node.

2. Router scenarios

Instead of store-and-forward routing (commonly employed in TCP/IP networks), an SpW router uses wormhole routing—a form of cut-through routing [49]—to transfer packets [13]. While a store-and-forward router buffers a packet in its entirety before forwarding it to the next node, a wormhole router forwards a packet as soon as the router finishes examining the packet's header to determine the output port to which the packet is forwarded. For SpaceWire, if the output port (link interface) is busy, the router stops receiving incoming packets until the output port becomes ready for transmission again. To inform the sending node of the "port not ready" condition, the router ceases sending flow control tokens (FCT) to the transmitting node. An FCT is a control character that an SpW link interface uses to tell its peer that it can accept eight more data characters[1].

While the output port is in the busy state, other packets that arrive on other SpW ports (on the same node) and are to be routed to the same output port will also be blocked until the output port is freed up. This could increase the transfer latency of the blocked packets since the SpW standard does not impose any limit on the maximum packet size. Employing *virtual channels* [50] over a single link to allow incoming packets to bypass blocked packets can ameliorate this problem. Nevertheless, without some form of temporal partitioning, the use of virtual channels alone may not be sufficient to avoid congestion. The SpW standard permits but does not mandate the use of virtual channels. Hence, we hypothesize that a timing channel could potentially exist in an SpW network that does not implement virtual channel flow control. A High node could leak information by varying the time it causes congestion at a particular port to signal to a Low node a value of 0 or 1. The Low node would receive the transmitted value by detecting a change in the transfer latency of its packets.

A wormhole router does not maintain I/O buffers because it is not required to buffer the entire packet before accepting the packet. Consequently, there is no shared buffer resource that could be exploited. However, the output ports are contentious shared resources. We speculate that a storage channel could be constructed using the FCT as a signaling mechanism. A High node could selectively create contention for a specific output port to send a 1 or 0 to a Low node. The Low node would try to send packets to the same port and could determine the value sent by the High node by observing whether the link interface connected to the Low node sends FCT(s) to the Low node or not.

[1] Multiple FCTs can be sent if there is space in the receiving link interface's receiver buffer for the additional data characters—one FCT for every eight spaces.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSION AND FUTURE WORK

This report describes the results of our study on covert channels that can potentially exist in 1553B and SpaceWire protocols. We have shown that it is possible to construct timing and storage channels by observing timing behavior, protocol control information, and implementation-specific features. Our approach is to find potential attack scenarios through analysis of protocol specifications and documentation of various products.

A. SUMMARY OF RESULTS

In our analysis of the 1553B bus protocol, we have identified three potential covert channels. A timing channel between a malicious High RT and a colluding Low RT could be constructed using the RT response time delay. Two storage channels could exist if the two cooperating RTs exploited the optional command illegalization functionality and the asynchronous data transfer mechanism.

For the SpaceWire analysis, we have surmised a potential timing channel in a mass memory device, and two channels (timing and storage) in a router. These channels arise from the use of shared resources in a multi-interface node by two conspiring external nodes.

B. FUTURE WORK

Two major tasks in the covert channel analysis process not addressed in this report are bandwidth estimation and mitigation of identified channels. For the identified 1335B channels, the next step would be an empirical study to estimate the rate at which information can be sent over each channel, and the feasibility and degree to which each channel can be exploited. Regarding mitigation, a brute force approach to handle the two storage channels is to disallow the use of command illegalization and asynchronous service request. However, doing so may not meet specific functional properties or absolute timing constraints of a particular mission. Alternatively, implementing an audit mechanism using a bus monitor may help detect and report suspicious behavior and potential exploits. The audit trail can be used to formulate a mitigating plan. A useful follow-up work is to confirm through testing that the identified channels are real channels and that the chosen the methods for handling the channels work as intended.

For SpaceWire, there are several directions for further research. One topic is to identify for potential storage channels that could be exploited via the SpW remote memory access protocol (RMAP) [51]. The SpW standard requires all routers to support an internal configuration port that can be used by applications to configure the routing logic. The SpW standard does not define a specific configuration mechanism but the RMAP is commonly used to read from and write to a router's configuration registers. The functionality and semantics of these registers are product-specific. Searching for these channels would require studying the configuration logic provided by different SpW products.

Another topic would be to determine if the Error End of Packet (EEP) condition could be used as a signaling mechanism. When a transmission error condition occurs, the SpW link will be disconnected and any packet that was being transferred will be terminated with a special EEP character. The data in an EEP-terminated packet can be valid and the EEP handling action depends on the type of the SpW component that receives the packet. An SpW router will transfer an EEP-terminated packet as a normal packet, i.e., a packet that ends with an End of Packet (EOP) marker. For a non-routing node, the EEP-terminated packet will be passed to the application level. This task requires studying vendor documentation to confirm the envisioned attack scenario(s).

Other error conditions that are reported to the network level (by the exchange level or packet level) are *link error* and *invalid destination address*. A link error will cause the link interface to be reset and, for a non-routing node, the error condition is reported to the application level. A link error may be flagged on a router for debugging and monitoring purposes. Further investigation is needed to determine if the link error condition could be exploited to leak information by measuring the time it takes to reset the link.

LIST OF REFERENCES

[1] U.S.-China Economic and Security Review Commission, "2011 Report to Congress of the U.S.-China Economic and Security Review Commission," November 2011.

[2] United States of America, Office of the President, "National Space Policy of the United States of America," June 28, 2010. Available at http://www.whitehouse.gov/sites/default/files/national_space_policy_6-28-10.pdf.

[3] United States Government Accountability Office, "2013 Annual Report: Reduce Fragmentation, Overlap, and Duplication and Achieve Other Financial Benefits," GAO-13-279SP, April 2013.

[4] Cueva, E. G., Esiely-Barrera, H. A., Kim, H. W. and Tang, Z., "Assessment of the Internet Protocol Routing in Space—Joint Capability Technology Demonstration," Johns Hopkins University Applied Physics Laboratory Technical Digest, Vol. 30, No. 2, 2011.

[5] Schueler, C., "Commercially-Hosted Payloads: Low-Cost Research to Operations," 934d American Meteorological Society Annual Meeting, Austin, TX, January 2013.

[6] U.S. Air Force, "Space and Missile Systems Center awards first-of-its-kind hosted payload solutions contract," July 28, 2014. Available at: http://www.af.mil/News/ArticleDisplay/tabid/223/Article/486870/space-and-missile-systems-center-awards-first-of-its-kind-hosted-payload-soluti.aspx.

[7] United States Committee on National Security Systems, "CNSSP No. 12 National Information Assurance Policy for Space Systems Used to Support National Security Missions," Government Standard, November 2012.

[8] Michael A. Padlipsky, David W. Snow, and Paul A. Karger, "Limitations of End-to-End Encryption in Secure Computer Networks," The MITRE Corporation: Bedford MA, HQ Electronic Systems Division technical report ESD-TR-78-158, August 1978.

[9] B. W. Lampson, "A Note on the Confinement Problem," Communications of the ACM, 16:10, pp. 613-615, October 1973.

[10] Millen, J., "20 years of covert channel modeling and analysis," Proceedings of the 1999 IEEE Symposium on Security and Privacy, pp.113-114, 1999.

[11] Military Standard MIL–STD–1553B: "Aircraft Internal Time Division Command/Response Multiplex Data Bus," September 21, 1978.

[12] Military Standard MIL–STD–1553B, Notice 1 – 4, February 1980 – January 1996.

[13] European Cooperation for Space Standardization, "ECSS-E-ST-50-12C SpaceWire - Links, nodes, routers and networks," July 2008.

[14] V. Gligor, "A Guide to Understanding Covert Channel Analysis of Trusted Systems," NCSC-TG-030, National Computer Security Center, November 1993.

[15] M. Schaefer, B. Gold, R. Linde, and J. Scheid, "Program Confinement in KVM/370," Proceedings of the 1977 Annual ACM Conference, Seattle, Washington, ACM, New York, pp. 404-410, October 1977.

[16] R. A. Kemmerer , “Shared Resource Matrix Methodology: An Approach to Identifying Storage and Timing Channels,” ACM Transactions on Computer Systems, 1 (3), pp. 256–277, August 1983.

[17] Chris deLong , “AS 15531/MIL-STD-1553B Digital Time Division Command/Response Multiplex Data Bus,” The Avionics Handbook, Ed. Cary R. Spitzer, Boca Raton, CRC Press LLC, 2001.

[18] Department of Defense Handbook MIL-HDBK-1553A: “Multiplex Applications Handbook,” November 1988.

[19] Department of Defense Handbook MIL-HDBK-1553A Notice 1 – 5, January 1993 – May 2013.

[20] Parkes, S., “SpaceWire User’s Guide,” STAR-Dundee Limited, 2012.

[21] Schueler, C. F., “FD-CHIRP: Hosted Payload System Engineering Lessons,” Orbital Sciences Corporation, August 2012.

[22] Armand, B., private communication, September 2014.

[23] United State Air Force, “Hosted Payload Standard Interface Specification (HPSIS),” Space and Missile Systems Center, Hosted Payload Office, , July 2013.

[24] United States Department of Defense, “DoD Instruction 8510.01, “DoD Information Assurance Certification and Accreditation Process (DIACAP),” Government Standard, November 28, 2007.

[25] United States Department of Defense, “DoD Instruction Number 8510.01 Risk Management Framework(RMF) for DoD Information Technology (IT),” Government Standard, May 2014.

[26] National Institute of Standards and Technology, “NIST Special Publication 800-37 Guide for Applying the Risk Management Framework to Federal Information Systems: A Security Life Cycle Approach,” February 2010, as amended.

[27] National Institute of Standards and Technology, “NIST Special Publication 800-53, Security and Privacy Controls for Federal Information Systems and Organizations,” Revision 4, April 2013.

[28] United States Committee on National Security Systems, “CNSSI No. 1200 National Information Assurance Instruction for Space Systems Used to Support National Security Missions,” Government Standard, May 2014.

[29] United States Committee on National Security Systems, “CNSSI No. 1253 Security Categorization and Control Selection for National Security Systems, Appendix F Attachment 2 Space Platform Overlay,” Government Standard, June 2013.

[30] United States Committee on National Security Systems, “CNSSI No. 1253 Security Categorization and Control Selection for National Security Systems, Appendix F Attachment 3 Cross Domain Solution (CDS) Overlay,” Government Standard, September 2013.

[31] National Institute of Standards and Technology, “NIST Special Publication 800-53, Security and Privacy Controls for Federal Information Systems and Organizations,” Revision 3 Errata 1, May 2010.

[32] Committee on National Security Systems. “Security Categorization and Control Selection for National Security Systems,” CNSS Instruction No. 1253, 15 March 2012.

[33] S. Zander, G. Armitage, and P. Branch, “A Survey of Covert Channels and Countermeasures in Computer Network Protocols,” IEEE Communications Surveys & Tutorials, IEEE, Volume:9, Issue: 3, July 2007.

[34] D. Bell and L. La Padula. “Secure Computer Systems: Unified Exposition and Multics Interpretation,” Electronic Systems Division, USAF. ESD-TR-75-306, MTR-2997 Rev.1. Hanscom AFB, MA. 1976.

[35] Millen, J., “Information Flow Analysis of Formal Specifications,” Proc. IEEE Symp. Security and Privacy, Apr. 1981, pp. 3–8.

[36] Richard A. Kemmerer, R. A., “A Practical Approach to Identifying Storage and Timing Channels: Twenty Years Later,” Proceedings of the18th Annual Computer Security Applications Conference (ACSAC '02), December 2002.

[37] J. A. Goguen and J. Meseguer, “Security Policies and Security Models,” Proceedings of the IEEE Symposium on Security and Privacy, pp. 11-20, April 1982.

[38] Linde, R. R., “Operating System Penetration,” in Proceedings of the National Computer Conference, pp. 361-367, 1975.

[39] Weissman, C., “Security Penetration Testing Guideline,” Naval Research Laboratory, Unisys Government Systems, 12010 Sunrise Vally Drive, Reston, VA, tm - 8889/000/01, October 1993. Prepared under contract to NRL.

[40] Weissman, C., “Penetration Testing,” in Abrams, Jajodia, and Podell, editors. Information Security: An Integrated Collection of Essays, pp. 269-296. IEEE Computer Society Press, Los Alamitos, CA, 1995.

[41] Nguyen, T. D., “Towards MIL-STD-1553B Covert Channel Analysis,” Naval Postgraduate School Technical Report NPS-CAG-15-001, January 2015.

[42] AIM GmbH, “MIL-STD-1553 Interface Module Programmer's Guide,” V22.9x Rev. A, May 2014.

[43] Alta Data Technologies LLC, “AltaAPI Software User’s Manual,” Rev I2, April 24, 2014.

[44] Excalibur Systems, Inc., “1553Px Family Software Tools Programmer’s Reference,” Rev C-4, July 2014.

[45] Microsemi Corporation, “Core1553BRM v4.0 Handbook,” January 2014.

[46] Aeroflex Colorado Springs, Inc., “Enhanced SμMMIT Family Hand Book,” May 2014.

[47] Garcia, J. I. et al., “Overview Of The INTAμSAT’S Mass Memory Unit Based On SpaceWire,” Proceedings of the 3rd International SpaceWire Conference, June 2010.

[48] ARM Limited, “AMBA Specification (Rev 2.0),” May 1999.

[49] Kermani, P. and Kleinrock, L., “Virtual Cut–Though: A New Computer Communication Switching Technique,” North Holland Publishing Company, Computer Networks 3 (1970) pp. 267-269.

[50] Dally, W.J., “Virtual-channel flow control,” Parallel and Distributed Systems, IEEE Transactions on , vol.3, no.2, pp.194,205, Mar 1992.

[51] European Cooperation for Space Standardization, “ECSS-E-ST-50-52C SpaceWire - Remote memory access protocol,” February 2010.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California

3. Research Sponsored Programs Office, Code 41
   Naval Postgraduate School
   Monterey, CA 93943